*ERD:*



*DB requirements:*

The DB that we want to create for this company will cover the online service requesting process and the online course registration. Therefore, our data base will keep tracks of the company Employees, the services that the company provides for organizations, and the courses that the company provides for individuals, also the organizations’ information and the trainees’ information who benefit from this company Also, it will keep tracks of the course payments that are made by the trainees. We don’t need to track the organization payments since it done after agreement and paper work, we will just store the price of requesting a service by an organization. So, the users of our data base are (Organizations, Trainees and Employees of our company).

* For each Employee, DB keeps track of unique employee identifier, name, address, and phone, his role (job) in the company, salary, and password.
* For each service, DB keeps track of unique service identifier, service name, service description (there are no more important characteristics other than these since the whole service depends on employee checking and paper work.
* For each request of a service, DB keeps track of unique request identifier, date of request and cost of conducting this specific service for this particular organization. The employee will visit the organization to check and do the work (service) according to that.
* For each organization requests a service, DB keeps track of unique organization identifier, organization name, and sector (private or public), location, phone, password.
* For each Trainee takes a course, DB keeps track of unique trainee identifier, name, and phone, password.
* For each course, DB keeps track of unique course identifier, course name, description, and course tuition, duration (number of weeks).
* For each payment by the trainee for a course, DB keeps track of unique card number, card type, and the billing address.
* Each service is requested via 0 or many request transactions, and each request transaction has exactly one service.
* Each organization requests via 0 or many request transactions. Each request belongs to exactly one organization.
* Each request transaction is assigned to 1 or many employees who work on it, and each employee works on 0 or many requests.
* Each trainee takes 0 or many courses, and each courses has 0 or many trainees. The same trainee can take the same course but in different year to refresh his certificate or to get one if he fails on the previous one (rule of the course maker).
* Each completion contains exactly one payment, and each payment can be included in many completion since the same trainee might be using the same card to pay for different completions.
* Each employee (who works as an instructor) teaches 0 or many courses, and each course is thought by exactly one employee since the company is small and have limited instructors.

*Relational Model:*

**

*DB Relations:*

We will have 9 relations in our database (from the relational model *–above-* we can see all of them and their columns, also the PK for each one of them).

*FD closure Test:*

All of our relations have full functional dependencies, no partial or transitive FD. There is no need to do any FD test, since all of them are in the 3NF. We have created our relational model from the ERD, which was based on the first description of our company. It was a straightforward process, which meant we didn’t need to normalize any of them.

*Error checking:*

In all interfaces, we check first if our user actually is registered or not (if he has a record in our Database or not) in this case we will make sure none of the database constrains will be affected and that the process will works smoothly without errors.